Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 623849, 8 pages
http://dx.doi.org/10.1155/2014/623849

Research Article

Synchronous Routing for Personal Rapid Transit Pods

Mehdi Mrad,[1] Olfa Chebbi,[2] Mohamed Labidi,[1] and Mohamed Ali Louly[1]

[1] *Department of Industrial Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia*
[2] *High Institute of Management, University of Tunis, 41 Rue de la Liberte, 2000 Le Bardo, Tunisia*

Correspondence should be addressed to Mohamed Labidi; mlabidi@ksu.edu.sa

Received 7 February 2014; Revised 3 June 2014; Accepted 6 June 2014; Published 26 June 2014

Academic Editor: M. Chadli

Personal rapid transit (PRT) is a public and automated transport system in which a fleet of small driverless vehicles operate in order to transport passengers between a set of stations through a network of guided ways. Each customer is carried from one station to another directly with no stop in intermediate stations. This mode of transport can result in a high level of unused capacity due to the empty moves of the vehicles. In this paper, we model the problem of minimizing the energy consumed by the PRT system while assuming predeterministic list of orders; then we solve it using some constructive heuristics. Experiments are run on 1320 randomly generated test problems with various sizes. Our algorithms are shown to give good results over large trip instances.

1. Introduction

Using public transportation means, such as train, metro, or bus, can lead to lengthy and suffering journey. The consecutive stops at the intermediate stations, the long waiting times at the stations, and the multiple line changing make the trip less comfortable. Using private cars seems to be a solution. However, the high cost related to the fuel consumption, pollution, and congestion problems presents serious drawbacks to private car users (see Donatos and Kioulafas [1] and Banister [2]).

In order to create a new transportation technology, an increasing effort is provided by researchers worldwide in different research areas including economy, energy, urban planning, safety, sustainable development, and transport environment (see Townsend and Zacharias [3] and Chadli et al. [4]).

In this context, the personal rapid transit (PRT) is considered as an interesting emerging technology. The basic idea of PRT can be traced back to 1953. The first academic article that treated the PRT system was written by Fichter [5]. And the first real implementation of a PRT system was made in the 70s by the University of West Virginia in Morgan town to connect the three-town campus as well as the down town area. In fact, PRT is an innovative system which combines the convenience of private travel and the advantage of public transport. PRT is composed generally of an electrical driverless fleet of vehicles called pods (Figure 1) that can take passengers directly from their departure station to their destination without any stop. This is possible due to the bypasses that exist in every station (Figure 2) which allow PRT pods to bypass the intermediate stations until arriving to their destination without stopping (see Won et al. [6] and Schweizer and Mantecchini [7]). A central automatic system guides the pods through a network of dedicated guides ways [8].

There was no real interest in the PRT system until the 90s when technology progresses related to the PRT system such as battery vehicle, stations structures, and automated control systems were made (see Dahmani et al. [9] and Dahmani et al. [10]). During the 90s, different PRT projects were prospected and studied. In fact, the taxi 2000 system [11] was selected for testing and construction in O'hare airport (Rosemont, USA). At present, there are a number of PRT systems that are operational as the Heathrow airport PRT system (London, UK) which allows people to travel from the car parking to Terminal 5 in the airport and the Masdar city (Abu Dhabi, UAE) which is planned to be the first zero carbon emission city where PRT is the only powered intercity transportation system.

Two routing approaches for the PRT pods exist. The first one is the synchronous (or clear path) approach where a list of scheduled trips is planned before pods leave the depot. In this

(a) (b)

Figure 1: PRT vehicle.

Figure 2: PRT station.

case, the central automatic system that guides the pods should prepare the route of each pod before starting the execution of the different trips [12]. The second one is the asynchronous approach where pods move from one station to another basically upon the request of customers. This approach seems more comfortable for customers but harder to control for the system supervisor. In this latter case, it becomes difficult to predict the congestion problems in the PRT network. The decision of the automatic system that guides the pods will be restricted to the assignment of the nearest available pod to the customer (see Kornhauser and McEvaddy [13] and Irving [14]).

A comparative study of the two approaches was made by Xithalis [15]. It is shown that despite the fact that the clear path approach offers less flexibility to customers, it is still much more interesting than the asynchronous approach because of its high capacity, congestion-free operation, and less topology limitations and also because it offers communications that are not real-time critical [8].

Unfortunately, both of routing approaches can result in a high level of unused capacity and wasted energy. However due to its deterministic aspect, the clear path approach offers the opportunity to optimize the wasted energy by designing optimal routing of the PRT pods based on the list of scheduled trips and the characteristics of the PRT network. In this context, many exact methods based on linear programming are presented for the vehicle routing problem with time window (see Baldacci et al. [16], Baños et al. [17], and Çetinkaya et al. [18]). These exact methods are time consuming procedures; therefore, some heuristic algorithms are presented; among them we quote briefly (Pang [19], Ursani et al. [20], and Belhaiza et al. [21]). The aim of this paper is to minimize the total consumed energy in the case of synchronous routing approach of PRT pods while ensuring all the scheduled trips.

This paper is organized as follows: Section 2 presents the problem definition and the mathematical formulation; Section 3 includes a description of the constructive heuristics proposed to solve the problem. We present the results of the proposed methods in Section 4. The conclusion follows.

2. Problem Definition and Mathematical Formulation

In this section, we present the problem definition as presented in the work of Mrad and Hidri [22]. Consider a connected network of personal rapid transit (PRT) with a finite number of stations (M) and one depot. We assume that we have a deterministic list of trips and that the vehicles have a limited battery capacity (B). Thus, the road from the depot to itself while visiting some stations must never use more than B electricity charge. We suppose also that we have an unlimited number of vehicles to serve all the passengers demands.

The set of trips has the cardinality $|T| = n$. Each trip i ($i = 1, \dots, n$) has a departure time Dt_i, a departure station Ds_i, an arrival time At_i, and an arrival station As_i. It is worth noting that the arrival time At_i is the departure time Dt_i plus the duration of the shortest path from the departure to the arrival stations. Furthermore, $\mathrm{Sp}_{(i,j)}$ denotes the lasted duration taking the shortest path from station i to station j.

In the sequel, we present an integer programming formulations, based on the following underlying network representation. Consider the graph $G = (V, E)$, with the set of nodes $V = T \cup \{s, t\}$, where s and t are two dummy nodes and E is the set of arcs. Denote $V^* = V/\{s, t\}$ and $E^* = E/\{(i, j); i = s \text{ or } j = t\}$, and sort the trips in the increasing order of their departure time. The set of arcs E is defined as follows.

(i) If $i, j \in V^*$ with $j > i$ and $\mathrm{At}_i + \mathrm{Sp}_{(\mathrm{As}_i, \mathrm{Ds}_j)} \leq \mathrm{Dt}_j$, then we add an arc (i, j) with cost c_{ij} representing the energy consumed from the arrival station of trip i (As_i) to the arrival station of trip j (As_j).

```
integer: GlobalCost ← 0, i ← 0
Procedure RelaxedProb(integer TotalCost[], integer PathStructure[][])

for (i < Size(TotalCost) do
  if (TotalCost[i] > B) then
    GlobalCost ← GlobalCost + GreedyCost(PathStructure[i])
  else
    GlobalCost ← GlobalCost + TotalCost[i]
  end if
end for
Return (GlobalCost)
```

Algorithm 1: Greedy correction.

```
integer GlobalCost ← 0
Procedure RelaxedProb(integer TotalCost[], integer PathStructure[][])

Function integer InitialProb(integer trip[], integer Cost[][])
for (j < Size(TotalCost) do
  if (TotalCost[j] > B) then
    integer trip[]
    for (i < Size(PathStructure[j]) do
      trip[i] ← PathStructure[j][i]
    end for
    GlobalCost ← ProbInitial (trip, Cost)
  else
    GlobalCost ← GlobalCost + TotalCost(j)
  end if
end for
Return (GlobalCost)
```

Algorithm 2: Optimal correction.

(ii) For each node i, we add an arc (s, i) (the cost of this arc is c_{si} and it represents the energy used to reach the arrival station of trip i, from the depot).

(iii) For each node i, we add an arc (i, t) (the cost of this arc is c_{it} and it represents the energy used from the arrival station of trip i to the depot).

2.1. An Assignment-Based Formulation. In this section, an assignment-based formulation is presented. To that aim, the following decision variables and definitions are introduced.

(i) $x_{ij} = 1$, if the station j is visited immediately after station i and 0, otherwise.

(ii) z_i is the amount of charge used to reach the node $i \in V^*$ from the depot.

(iii) Consider $a_i = c_{si}$ for $i \in V^*$.

(iv) Consider $b_i = B - c_{it}$ for $i \in V^*$.

In addition, we define the following:

(i) $\delta^+(i)$ is the set of nodes j such that an arc (i, j) exists;

(ii) $\delta^-(i)$ is the set of nodes j such that an arc (j, i) exists.

Hence, the minimum charge assuring the trips is equal to the optimal value of the following programming model:

$$\textbf{PRT (1)}: \quad \text{Minimize} \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{1}$$

$$\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i \in V^* \tag{2}$$

$$\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall i \in V^* \tag{3}$$

$$z_i + c_{ij} \le z_j + \left(b_i - a_j + c_{ij}\right)\left(1 - x_{ij}\right) \quad \forall (i, j) \in E^* \tag{4}$$

$$a_i \le z_i \le b_i \quad \forall i \in V^* \tag{5}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \tag{6}$$

$$z_i \ge 0 \quad \forall i \in V^*. \tag{7}$$

Objective (1) is to minimize the total used charge. Constraints (2) and (3) require that each node $i \in V^*$ must be visited and left only one time, respectively. Constraints (4) ensure the following conditions.

(i) If $x_{ij} = 1$, then $z_i + c_{ij} \le z_j$ with $a_i \le z_i \le b_i$ $\forall i, j \in V^*$.

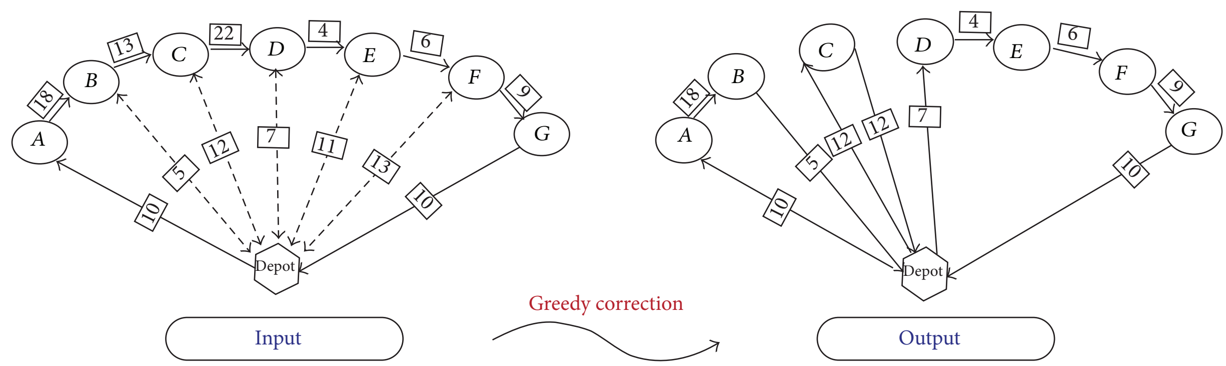

Figure 3: Illustration of greedy correction heuristic.

Constraints (5) present a trivial bound limitation on the charge needed to perform the trip i. Finally, constraints (6) and (7) indicate that the decision variables x_{ij} are binary-valued and z_i are positive real variables.

3. Constructive Methods

In this section, we present three constructive heuristics for the problem considered above.

We first solve the following relaxed linear program:

$$\begin{aligned}
\textbf{RLPRT}:\quad & \text{Minimiser} \sum_{(i,j)\in E} c_{ij}x_{ij} \\
& \sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i \in V^* \\
& \sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall i \in V^* \\
& x_{ij} \in \{0,1\} \quad \forall (i,j) \in E.
\end{aligned} \tag{8}$$

However, by solving this linear program, we can get infeasible tour which consumes more energy than what the battery allows. To fix this dilemma, we propose three different algorithms which are presented hereafter.

3.1. Greedy Correction Algorithm

3.1.1. Approach. Each infeasible path can be divided into a set of feasible paths using the following method. Let S_j be the sequence of trips that should be visited by a PRT car in the jth infeasible path. We start by the first trip in the sequence S_j and we continue to add trips while the cumulative energy required by these trips is less or equal to the battery capacity. Once the battery capacity is exceeded, a new path should be considered starting from the first noncovered node in the sequence S_j. We continue to build feasible paths till all trips of S_j are covered. This method will be applied on all infeasible paths.

3.1.2. Illustration. It is supposed that after the resolution of the relaxed linear program, which does not consider the battery's capacity, we will find a set of infeasible paths. Each infeasible path begins and returns to the depot in order to make trips $A, B, \ldots, G$ Figure 3 includes an example illustrating the correction of infeasible routes based on the heuristic described above. The capacity of the battery in this example is equal to 40.

3.1.3. Algorithm. Algorithm 1 describes in detail the different steps of the heuristic. We suppose that we have the procedure RelaxedProb (integer TotalCost [] and integer PathStructure [][]) which takes two parameters: an empty array TotalCost which is filled by the total cost of each path provided using this method. The second parameter is the matrix PathStructure which will be completed by all the different paths of itinerary. We suppose also that the function size (array) takes an array as an argument and returns its size. The function GreedyCost should take the structure of an infeasible path and try to correct it in order to respect the battery's capacity. It will split the infeasible route to a set of feasible routes.

3.2. Split Correction Algorithm

3.2.1. Approach. The route-first cluster second approach was introduced by Beasley [23]. The main idea of this approach consists of building an auxiliary graph, with the vertices of an infeasible path including the depot. An arc (i, j) such that $i < j$ in the graph represents a feasible tour from i to j which visits the vertices $i + 1, i + 2, \ldots, j$. The final extremity of an arc represents the end of the tour, and the cost of the arc is the cost of the tour. Only arcs having cost less than the capacity of the battery are considered in the graph. An optimal partition is obtained by solving the shortest path problem on the obtained auxiliary graph. Each arc (i, j) considered in the solution of the shortest path problem represents a feasible tour including trips $(i, i + 1, \ldots, j)$ that should be considered in the solution of the PRT problem.

3.2.2. Illustration. Figure 4 represents an application of the Split procedure to get the optimal splitting of a tour for the VRP problem. Figure 4(a) shows a sequence $S = (a, b, c, d, e)$ and the costs are written on the arcs. The middle graph

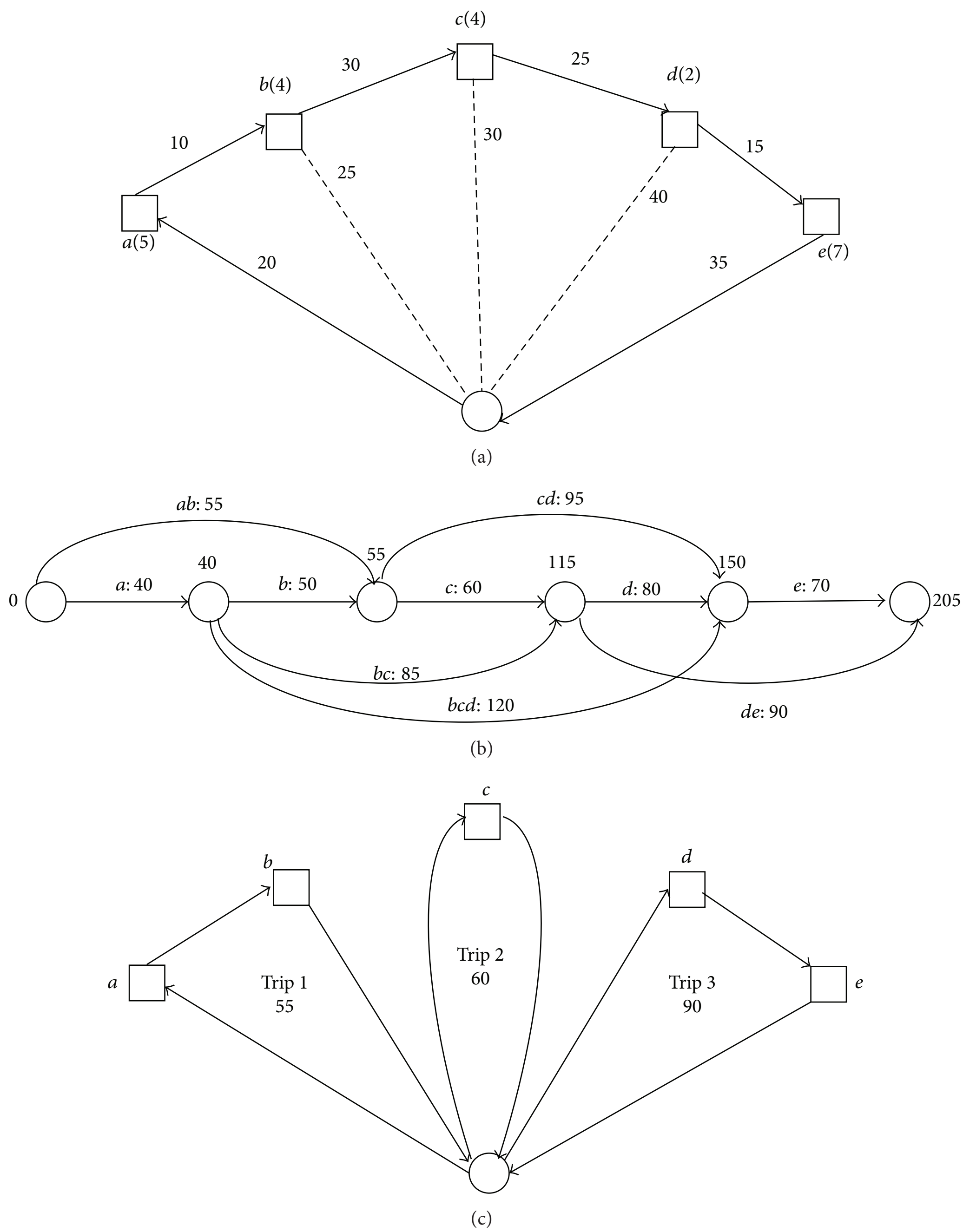

FIGURE 4: Example of Split procedure [18].

contains such an arc ab with a cost of 55 for the trip $(0, a, b, 0)$. The bold path has three arcs, and its cost is 255. The lower part gives the resulting solution with three trips.

3.3. Optimal Correction Algorithm

3.3.1. Approach. Solving the relaxed PRT problem (RLPRT) will lead to a set of paths. Unfortunately, it is possible that some infeasible paths may exist in this set. In this heuristic, we will correct the infeasible resulting tours while considering that the set of trips included in an infeasible path is in itself a new subproblem that can be solved optimally using the formulation (PRT(1)). In fact, assuming that the number of trips that belong to only one path is reduced, then the resulting subproblem can be solved to optimality in a short time using integer linear programming.

3.3.2. Illustration. Figure 5 illustrates how a feasible solution can be found through two steps. In the first step, the resolution of the relaxed integer linear program (RLPRT)

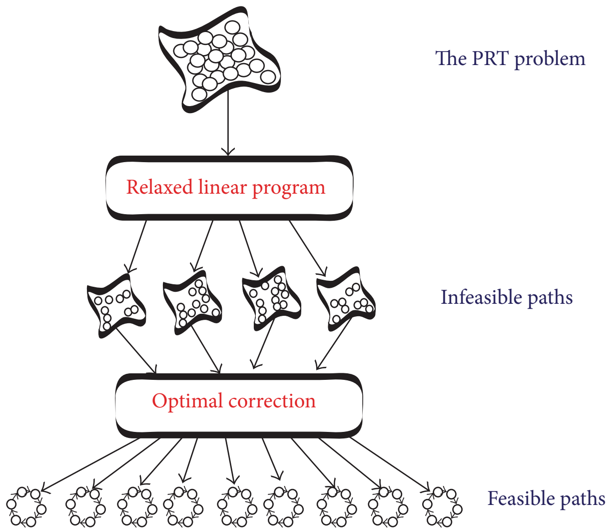

FIGURE 5: Illustration of optimal correction heuristic.

TABLE 1: Instance assumptions.

Number of stations	12
Cost of arcs	Generated randomly between 1 and 15
Departure station	Generated randomly between 1 and 12
Arrival station	Generated randomly between 1 and 12
Departure time	Generated randomly between 1 and 3600
Arrival time	Sum of the departure time and the trip duration
Battery capacity	40 minutes
Max-wait	1000 seconds

will provide a set of paths potentially infeasible because no constraint related to the battery capacity is considered. In the second step, each set of trips belonging to an infeasible path will be considered as the input of the integer linear program PRT(1) which ensures the feasibility of any solution regarding the requirement of the PRT system as described in this paper.

3.3.3. Algorithm. Algorithm 2 describes the process of solving the problem using two main functions. The first RelaxedProb (integer TotalCost [] and integer PathStructure [][]) solves the relaxed integer linear program (RLPRT). The second function InitialProb (integer trip [] and integer Cost [][]) solves the original linear program PRT(1) which takes into account the constraints of the battery. This function returns the total cost of the different tours built from the set of trips of the corrected infeasible path.

4. Computational Results

All the algorithms were coded in C++ and compiled with Visual Studio 2010. The LP instances have been solved using Cplex 12.2. All the computational experiments were carried out on a Dual Core 2.70 GHz Personal Computer with 1 GB RAM under Windows XP.

TABLE 2: Results of the greedy correction heuristic.

Size	Average gap (%)	Average time (sec)
10	3.491	0.044
15	3.880	0.033
20	3.777	0.023
25	6.104	0.028
30	7.408	0.027
35	7.222	0.037
40	6.354	0.040
45	5.768	0.042
50	7.400	0.053
55	7.605	0.174
60	7.568	0.082
65	7.046	0.170
70	7.671	0.085
75	8.243	0.150
80	9.669	0.178
85	8.109	0.294
90	7.901	0.183
95	8.148	0.121
100	8.148	0.120
110	8.361	0.137
120	8.478	0.169
130	8.758	0.181
140	9.973	0.193
150	11.083	0.301
160	8.347	0.224
170	9.410	0.237
180	9.424	0.231
190	10.613	0.287
200	9.498	0.304
250	8.617	0.613
300	12.005	0.879
350	11.373	1.182
400	13.603	1.685
Average	8.214	0.258

4.1. Data Sets. We generated 40 instances for 33 values of n ranging in $[10, 400]$. So, in general we did test the proposed heuristics on 1320 different instances. Table 1 summarizes the generating principle of each type of instances.

In order to evaluate the results, we computed the average relative gap (GAP) defined by

$$\text{GAP} = \left(\frac{S_{\text{Heuristic}} - \text{LB}}{\text{LB}}\right) * 100. \tag{9}$$

(i) $S_{\text{Heuristic}}$ is the solution of the heuristic.

(ii) LB is the lower bound obtained by the linear relaxation of PRT(1) (see Section 2).

4.2. Results. The performance of all the developed heuristics are shown in Tables 2, 3, and 4. We could note that the best

Table 3: Results of the split correction heuristic.

Size	Average gap (%)	Average time (sec)
10	2.529	0.022
15	2.701	0.018
20	2.476	0.018
25	4.055	0.021
30	5.536	0.017
35	5.544	0.020
40	4.596	0.027
45	4.199	0.025
50	5.464	0.030
55	5.774	0.035
60	5.642	0.243
65	4.782	0.195
70	5.727	0.107
75	5.591	0.187
80	7.158	0.114
85	5.911	0.089
90	5.561	0.212
95	5.717	0.097
100	6.036	0.125
110	6.229	0.127
120	6.056	0.153
130	6.348	0.211
140	7.198	0.609
150	8.499	0.216
160	6.330	0.165
170	6.869	0.173
180	6.799	0.266
190	7.829	0.267
200	6.692	0.297
250	6.188	0.450
300	8.882	0.808
350	8.465	1.065
400	10.605	1.479
Average	6.000	0.239

Table 4: Results of the optimal correction heuristic.

Size	Average gap (%)	Average time (sec)
10	2.529	0.051
15	2.603	0.034
20	2.436	0.047
25	4.005	0.055
30	5.174	0.068
35	5.475	0.095
40	4.566	0.089
45	4.169	0.106
50	5.318	0.155
55	5.663	0.154
60	5.406	0.201
65	4.757	0.152
70	5.634	0.168
75	5.470	0.200
80	7.008	0.277
85	5.828	0.242
90	5.447	0.262
95	5.594	0.254
100	5.936	0.282
110	6.162	0.284
120	5.956	0.402
130	6.252	0.300
140	7.056	0.420
150	8.359	0.532
160	6.260	0.567
170	6.809	0.619
180	6.718	0.658
190	7.718	0.691
200	6.637	0.791
250	6.112	1.113
300	8.802	1.610
350	8.383	2.285
400	10.503	2.987
Average	5.901	0.489

heuristic in terms of average gap is the optimal correction heuristic with a value of 5.901% and in terms of computational time the best one is the split correction heuristic which has an average time of 0.239 seconds.

5. Conclusion

Given the ecological, economic, and social interests, the personal rapid transit (PRT) is a research area that attracts more and more researchers from different domains. Optimizing the energy used by a PRT system is an important and challenging problem that has a direct impact on the cost of this new transportation mean. Since the exact approaches are time consuming and are not able to solve large instances (see Mrad and Hidri [22]), we propose in this paper to solve heuristically larger instances. In this work, three different constructive heuristics were presented. Our methods can find a high quality solution for almost all the different randomly generated instances. As an extension to this work, metaheuristics can be developed and other versions of the PRT problem can be considered by changing the charging strategy of the batteries or by limiting the fleet size of PRT vehicles.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-296.

References

[1] G. S. Donatos and K. E. Kioulafas, "A quantitative analysis of new car sales and advertising in Greece," *European Journal of Operational Research*, vol. 48, no. 3, pp. 311–317, 1990.

[2] D. Banister, "Cities, mobility and climate change," *Journal of Transport Geography*, vol. 19, no. 6, pp. 1538–1546, 2011.

[3] C. Townsend and J. Zacharias, "Built environment and pedestrian behavior at rail rapid transit stations in Bangkok," *Transportation*, vol. 37, no. 2, pp. 317–330, 2010.

[4] M. Chadli, S. Aouaouda, H. R. Karimi, and P. Shi, "Robust fault tolerant tracking controller design for a VTOL aircraft," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2627–2645, 2013.

[5] D. Fichter, *Individualized Automatic Transit and the City*, 1964.

[6] J. Won, H. Choe, and F. Karray, "Optimal design of personal rapid transit," in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC '06)*, pp. 1489–1494, Toronto, Canada, September 2006.

[7] J. Schweizer and L. Mantecchini, *Performance Analysis of Large Scaleprt Networks: Theoretical Capacity and Microsimulations*, Bologna University, DISTART Trasporti.

[8] J. E. Anderson, *Transit Systems Theory*, Lexington Books, Lexington, Ky, USA, 1978.

[9] H. Dahmani, M. Chadli, A. Rabhi, and A. El Hajjaji, "Road curvature estimation for vehicle lane departure detection using a robust Takagi-Sugeno fuzzy observer," *Vehicle System Dynamics*, vol. 51, no. 5, pp. 581–599, 2013.

[10] H. Dahmani, M. Chadli, A. Rabhi, and A. El Hajjaji, "Vehicle dynamic estimation with road bank angle consideration for rollover detection: theoretical and experimental studies," *Vehicle System Dynamics*, vol. 51, no. 12, pp. 1853–1871, 2013.

[11] J. E. Anderson, "The taxi 2000 personal rapid transit system," *Journal of Advanced Transportation*, vol. 22, no. 1, pp. 1–15, 1988.

[12] J. E. Anderson, "Control of personal rapid transit systems," *Journal of Advanced Transportation*, vol. 32, no. 1, pp. 57–74, 1998.

[13] A. L. Kornhauser and P. McEvaddy, "A quantitative analysis of synchronous vs. quasi-synchronous network operations of automated transit systems," *Transportation Research*, vol. 9, no. 4, pp. 241–248, 1975.

[14] J. H. Irving, *Fundamentals of Personal Rapid Transit: Based on a Program of Research, 1968–1976 at the Aerospace Corporation*, Lexington Books, El Segundo, Calif, USA, 1978.

[15] C. Xithalis, "Synchronous control method for personal rapid transit systems," 2008.

[16] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[17] R. Baños, J. Ortega, C. Gil, A. L. Márquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 286–296, 2013.

[18] C. Çetinkaya, I. Karaoglan, and H. Gökçen, "Two-stage vehicle routing problem with arc time windows: a mixed integer programming formulation and a heuristic approach," *European Journal of Operational Research*, vol. 230, no. 3, pp. 539–550, 2013.

[19] K. W. Pang, "An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11939–11946, 2011.

[20] Z. Ursani, D. Essam, D. Cornforth, and R. Stocker, "Localized genetic algorithm for vehicle routing problem with time windows," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5375–5390, 2011.

[21] S. Belhaiza, P. Hansen, and G. Laporte, "A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows," *Computers and Operations Research*, 2013.

[22] M. Mrad and L. Hidri, "Optimal consumed electric energy while sequencing vehicle trips in a personal rapid transit transportation system," *Computers and Industrial Engineering*. In press.

[23] J. Beasley, "Route first-cluster second methods for vehicle routing," *Omega*, vol. 11, no. 4, pp. 403–408, 1983.